



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ

«Информатика и системы управления»

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 16

Студент: Керимов А. Ш.

Группа: ИУ7-64Б

Преподаватель: Толпинская Н. Б.

Москва.
2020 г.

Цель работы — изучить использование правил в программе: структуру, особенности оформления, а также, способ и принципы выполнения таких программ на Prolog.

Задание.

Ответы на вопросы:

- **В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)**

Пролог выполняет унификацию в двух случаях: когда цель сопоставляется с заголовком предложения или когда используется знак равенства, который является инфиксным предикатом (предикатом, который расположен между своими аргументами, а не перед ними).

- **Каковы назначение и результат использования алгоритма унификации?**

Алгоритм унификации необходим для попытки "увидеть одинаковость" — сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией. Результат унификации — ответ «да» или «нет».

- **Какое первое состояние резольвенты?**

Вопрос

- **Как меняется резольвента?**

Резольвента - текущая цель, существующая на любой стадии вычислений. Резольвенты порождаются целью и каким-либо правилом или фактом, которые просматриваются последовательно сверху вниз. Если резольвента существует при наиболее общей унификации, она вычисляется. Если пустая резольвента с помощью такой стратегии не найдена, то ответ на вопрос отрицателен.

- **В каких пределах программы уникальны переменные?**

Именованные переменные уникальны в пределах одного предложения, анонимные уникальны все.

- **Как применяется подстановка, полученная с помощью алгоритма унификации?**

Подстановка применяется к подцелям резольвенты, путем конкретизации переменных.

- **В каких случаях запускается механизм отката?**

Механизм отката запускается в 2 случаях:

1. Если алгоритм попал в тупиковую ситуацию.
2. Если резольвента не пуста и решение найдено, но в базе знания остались не отмеченные предложения.

Создать базу знаний: «ПРЕДКИ», позволяющую наиболее эффективным способом (за меньшее количество шагов, что обеспечивается меньшим количеством предложений БЗ - правил), используя разные варианты (примеры) одного вопроса, определить (указать: какой вопрос для какого варианта):

1. по имени субъекта определить всех его бабушек (предки 2-го колена),
2. по имени субъекта определить всех его дедушек (предки 2-го колена),
3. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),
4. по имени субъекта определить его бабушку по материнской линии (предки 2-го колена),
5. по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).

Минимизировать количество правил и количество вариантов вопросов.

Использовать **конъюнктивные правила и простой вопрос**.

Для **одного** из вариантов **ВОПРОСА** и конкретной БЗ **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями:

очередная проблема на каждом шаге и метод ее решения;

каково новое текущее состояние резольвенты, как получено;

какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов?

Почему этих?);

вывод по результатам очередного шага и дальнейшие действия.

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1...	Комментарий, вывод...
...

Практическая часть

Листинг 1. Предки

```
domains
    name = symbol.

predicates
    father(name, name)
    mother(name, name)
    ancestor(name, name, name, name, name)

clauses
    father(azamat, ashur).
    father(evgeniy, aleksandr).
    father(aleksandr, nikolay).
    father(viktoria, enokentiy).
    father(ashur, petya).
    father(ekaterina, anton).

    mother(azamat, ekaterina).
    mother(evgeniy, viktoria).
    mother(ashur, tanya).
    mother(viktoria, anzhela).
    mother(aleksandr, avdotia).
    mother(ekaterina, natasha).

    ancestor(Name, GrandMotherM, GrandFatherM, GrandMotherF, GrandFatherF) :-
        mother(Name, Mother), mother(Mother, GrandMotherM), father(Mother, GrandFatherM),
        father(Name, Father), mother(Father, GrandMotherF), father(Father, GrandFatherF).

goal
    ancestor(azamat, GM, GF, GMM, GFM).
```

Результаты работы:

№	Вопрос	Ответ
1	ancestor(azamat, GMM, _, GMF, _).	GMM=natasha, GMF=tanya 1 Solution
2	ancestor(azamat, _, GMF, _, GFF).	GMF=anton, GFF=petya 1 Solution
3	ancestor(azamat, GMM, GMF, GFM, GFF).	GMM=natasha, GMF=anton, GFM=tanya, GFF=petya 1 Solution
4	ancestor(azamat, GMM, _, _, _).	GMM=natasha 1 Solution
5	ancestor(azamat, GMM, GMF, _, _).	GMM=natasha, GMF=anton 1 Solution

Сравнение происходит между подцелью из резольвенты и каждого знания из базы знаний, причем при каждой неудачной унификации запускается новый алгоритм унификации.

шаг	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	Резольвента {ancestor(azamat, GMM, GFM, GMF, GFF)}. Дальше – унификация. (Потому что резольвента не пуста)	Сравниваются ancestor(azamat, GMM, GFM, GMF, GFF) и ancestor(Name, GrandMotherM, GrandFatherM, GrandMotherF, GrandFatherF) Результат: Термы унифицируемые. Подстановка: {Name=azamat, GrandMotherM=GMM, GrandFatherM=GFM, GrandMotherF=GMF, GrandFatherF=GFF}	Прямой ход. (термы унифицируемы.) Содержимое резольвенты заменяется телом найденного правила - редукция. К резольвенте применяется подстановка.
2	Резольвента {mother(azamat, Mother), mother(Mother, GMM), father(Mother, GFM), father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Сравниваются mother(azamat, Mother) и mother(azamat, ekaterina). Термы унифицируемы. Подстановка: {azamat=azamat, Mother=ekaterina}	Прямой ход. (термы унифицируемы.) К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.
3	Резольвента {mother(ekaterina, GMM), father(ekaterina, GFM),	Сравнивается mother(ekaterina, GMM) и mother(ekaterina,	Прямой ход. (термы унифицируемы.) К резольвенте применяется

	father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	natasha). Термы унифицируемы. Подстановка: {GMM=natasha}	подстановка. Резольвента преобразуется с помощью редукции.
4	Резольвента {father(ekaterina, GFM), father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Сравнивается father(ekaterina, GFM) с father(ekaterina, anton). Термы унифицируемы. Подстановка: {GFM}	Прямой ход. (термы унифицируемы.)К резольvente применяется подстановка. Резольвента преобразуется с помощью редукции.
5	Резольвента {father(azamat, Father) , mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Сравниваются father(azamat, Father) с father(azamat, ashur). Термы унифицируемы. Подстановка: {Father=ashur}	Прямой ход. (термы унифицируемы.)К резольvente применяется подстановка. Резольвента преобразуется с помощью редукции.
6	Резольвента {mother(ashur, GMF) , father(ashur, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Сравнивается mother(ashur, GMF) с mother(ashur, tanya). Термы унифицируемы. Подстановка {GMF=Tanya}	Прямой ход. (термы унифицируемы.)К резольvente применяется подстановка. Резольвента преобразуется с помощью редукции.
7	Резольвента {father(ashur, GFF) } Дальше – унификация. (Потому что резольвента не пуста)	Сравнивается father(ashur, GFF) с father(ashur, petya). Термы унифицируемы. Подстановка {GFF=petya}	Прямой ход. (термы унифицируемы.)К резольvente применяется подстановка. Резольвента преобразуется с помощью редукции.
8	Резольвента { } (резольвента пуста все переменные связаны)		Вывод результата и поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.
9	Возврат предыдущего состояния резольвенты {father(ashur, GFF) } Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации Подцели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Откат к предыдущему состоянию резольвенты
10	Возврат предыдущего состояния резольвенты { mother(ashur, GMF) , father(ashur, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации Подцели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Откат к предыдущему состоянию резольвенты
11	Возврат предыдущего состояния резольвенты {father(azamat, Father) , mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации Подцели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Откат к предыдущему состоянию резольвенты
12	Возврат предыдущего состояния резольвенты	Происходят унификации Подцели из резольвенты	Унификации безуспешны. Откат к предыдущему

	{father(ekaterina, GFM), father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	и еще неотмеченных фактов из базы знаний.	состоянию резольвенты
13	Возврат предыдущего состояния резольвенты {mother(ekaterina, GMM), father(ekaterina, GFM), father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации Подцели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Откат к предыдущему состоянию резольвенты
14	Возврат предыдущего состояния резольвенты {mother(azamat, Mother) , mother(Mother, GMM), father(Mother, GFM), father(azamat, Father), mother(Father, GMF), father(Father, GFF).} Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации Подцели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Откат к предыдущему состоянию резольвенты
15	Резольвента {ancestor(azamat, GMM, GFM, GMF, GFF)}. Дальше – унификация. (Потому что резольвента не пуста)	Происходят унификации цели из резольвенты и еще неотмеченных фактов из базы знаний.	Унификации безуспешны. Завершение выполнения программы.